Deep Learning Concepts

Sergey Ivanov (617)

qbrick@mail.ru

September 16, 2019

1 Backpropagation
- Putting some pieces together
- Vector differentiation
- Backpropagation

Backpropagation

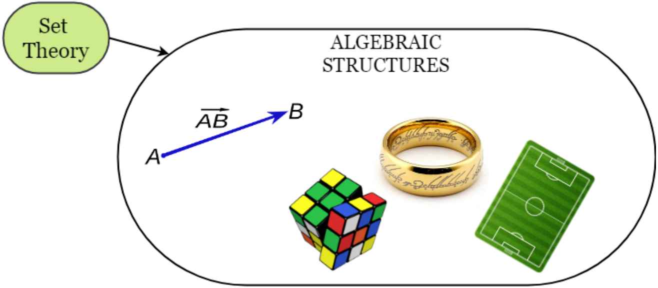
Putting some pieces together

Motivation to discuss again

- to have another view on vector differentiation
- to draw some connections between different subjects
- highlight theory we (implicitly?) utilize

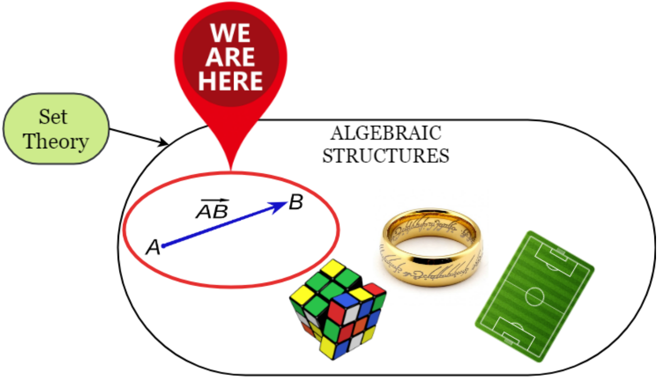
Motivation to discuss again

- to have another view on vector differentiation
- to draw some connections between different subjects
- highlight theory we (implicitly?) utilize



Motivation to discuss again

- to have another view on vector differentiation
- to draw some connections between different subjects
- highlight theory we (implicitly?) utilize

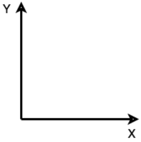


Finite vector spaces

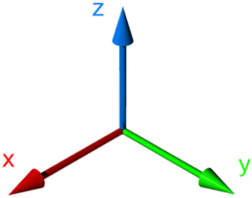
Theorem

All n-dimensional vector spaces[1] are isomorphic

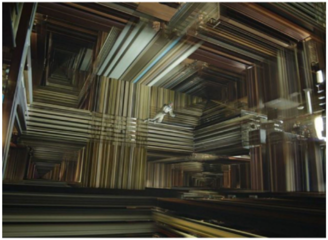
$\mathbb{R}^2$



$\mathbb{R}^3$



$\mathbb{R}^4 (\mathbb{R}^5 ?)$



[1] over same field (in our case — $\mathbb{R}$)

Key task!

$$f(x) : \mathbb{R}^n \to \mathbb{R}$$

$$f(x) \to \min_x$$

Key task!

$$f(x) : \mathbb{R}^n \to \mathbb{R}$$

$$f(x) \to \min_x$$

Alternative view:

How can we for some x_0 find x so that $f(x) < f(x_0)$?

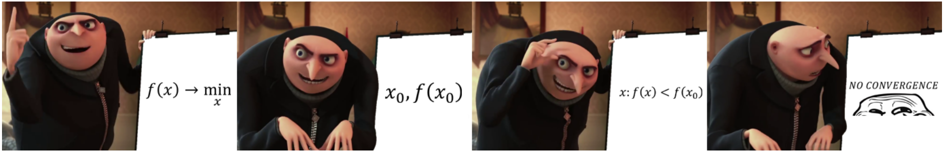
Key task!

$$f(x) : \mathbb{R}^n \to \mathbb{R}$$

$$f(x) \to \min_x$$

Alternative view:

How can we for some x_0 find x so that $f(x) < f(x_0)$?



Optimization step concept

Idea:

Let $f(x) = g(x) + h(x)$, where:

- $g(x)$ is something simple that can be easily optimized
- $h(x)$ is something that we can neglect

Optimization step concept

Idea:

Let $f(x) = g(x) + h(x)$, where:

- $g(x)$ is something simple that can be easily optimized
- $h(x)$ is something that we can neglect

What simple class of functions $g(x)$ to consider?

(a) $g(x+y) = g(x) + g(y) \quad \forall x, y \in \mathbb{R}^n$

Optimization step concept

Idea:

Let $f(x) = g(x) + h(x)$, where:

- $g(x)$ is something simple that can be easily optimized
- $h(x)$ is something that we can neglect

What simple class of functions $g(x)$ to consider?

(a) $g(x+y) = g(x) + g(y) \quad \forall x, y \in \mathbb{R}^n$

× some are discontinuous

Optimization step concept

Idea:

Let $f(x) = g(x) + h(x)$, where:

- $g(x)$ is something simple that can be easily optimized
- $h(x)$ is something that we can neglect

What simple class of functions $g(x)$ to consider?

(a) $g(x+y) = g(x) + g(y) \quad \forall x, y \in \mathbb{R}^n$

  × some are discontinuous

(b) $g(\alpha x) = \alpha g(x) \quad \forall \alpha \in \mathbb{R}, \forall x \in \mathbb{R}^n$

Optimization step concept

Idea:

Let $f(x) = g(x) + h(x)$, where:

- $g(x)$ is something simple that can be easily optimized
- $h(x)$ is something that we can neglect

What simple class of functions $g(x)$ to consider?

(a) $g(x + y) = g(x) + g(y) \quad \forall x, y \in \mathbb{R}^n$

$\times$ some are discontinuous

(b) $g(\alpha x) = \alpha g(x) \quad \forall \alpha \in \mathbb{R}, \forall x \in \mathbb{R}^n$

$\times$ some are discontinuous ($n > 1$)

Optimization step concept

Idea:

Let $f(x) = g(x) + h(x)$, where:

- $g(x)$ is something simple that can be easily optimized
- $h(x)$ is something that we can neglect

What simple class of functions $g(x)$ to consider?

(a) $g(x+y) = g(x) + g(y) \quad \forall x, y \in \mathbb{R}^n$

$\times$ some are discontinuous

(b) $g(\alpha x) = \alpha g(x) \quad \forall \alpha \in \mathbb{R}, \forall x \in \mathbb{R}^n$

$\times$ some are discontinuous ($n > 1$)

Consider (a) + (b) and everything will work out!

Linear functions

$$g : \mathbb{R}^n \to \mathbb{R}$$

$$g(x + y) = g(x) + g(y)$$

$$g(\alpha x) = \alpha g(x)$$

Question: How this class of functions can be described?

Linear functions

$$g : \mathbb{R}^n \to \mathbb{R}$$

$$g(x + y) = g(x) + g(y)$$

$$g(\alpha x) = \alpha g(x)$$

Question: How this class of functions can be described?

- $n = 1$:

Linear functions

$$g : \mathbb{R}^n \to \mathbb{R}$$

$$g(x + y) = g(x) + g(y)$$

$$g(\alpha x) = \alpha g(x)$$

Question: How this class of functions can be described?

- $n = 1$: $g(x) = kx$ for some $k \in \mathbb{R}$

Linear functions

$$g : \mathbb{R}^n \to \mathbb{R}$$

$$g(x + y) = g(x) + g(y)$$

$$g(\alpha x) = \alpha g(x)$$

Question: How this class of functions can be described?

- $n = 1$: $g(x) = kx$ for some $k \in \mathbb{R}$
  - **Proof:** $g(x) = g(x \cdot 1) = xg(1) = \{k := g(1)\} = kx$

Linear functions

$$g : \mathbb{R}^n \to \mathbb{R}$$

$$g(x + y) = g(x) + g(y)$$

$$g(\alpha x) = \alpha g(x)$$

Question: How this class of functions can be described?

- $n = 1$: $g(x) = kx$ for some $k \in \mathbb{R}$
  - **Proof:** $g(x) = g(x \cdot 1) = xg(1) = \{k := g(1)\} = kx$
- $n \geq 1$:

Linear functions

$$g : \mathbb{R}^n \to \mathbb{R}$$

$$g(x + y) = g(x) + g(y)$$

$$g(\alpha x) = \alpha g(x)$$

Question: How this class of functions can be described?

- $n = 1$: $g(x) = kx$ for some $k \in \mathbb{R}$
  - **Proof:** $g(x) = g(x \cdot 1) = xg(1) = \{k := g(1)\} = kx$
- $n \geq 1$: Riesz Representation Theorem

Riesz Representation Theorem

Riesz Theorem[2] (for finite vector spaces)

Every linear function $g : \mathbb{R}^n \to \mathbb{R}$ can be represented as $g(x) = \sum_i^n x_i y_i$ for some $y \in \mathbb{R}^n$

[2] proof is relatively simple

Riesz Representation Theorem

Riesz Theorem[2] (for finite vector spaces)

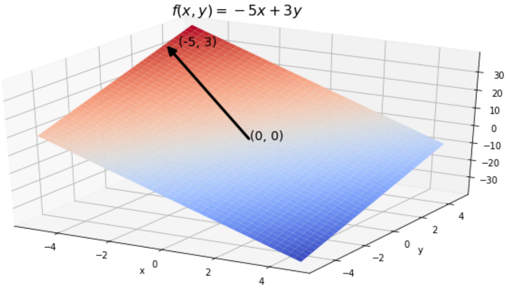
Every linear function $g : \mathbb{R}^n \to \mathbb{R}$ can be represented as $g(x) = \sum_i^n x_i y_i$ for some $y \in \mathbb{R}^n$

[2] proof is relatively simple

Riesz Representation Theorem

Riesz Theorem[2] (for finite vector spaces)

Every linear function $g : \mathbb{R}^n \to \mathbb{R}$ can be represented as $g(x) = \sum_i^n x_i y_i$ for some $y \in \mathbb{R}^n$



[2] proof is relatively simple

Linearization

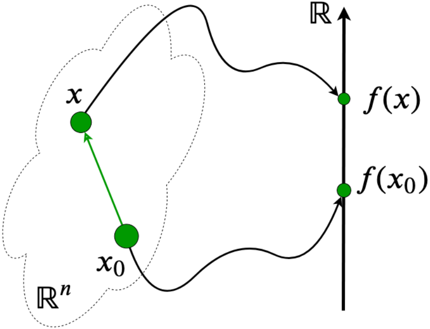
Let $x_0 \in \mathbb{R}^n$ be given point.

$$\underbrace{f(x) - f(x_0)}_{\text{change in function}} = \underbrace{g(x - x_0)}_{\substack{\text{linear part}\\\text{(differential)}}} + \underbrace{h(x - x_0)}_{\substack{\text{approximation}\\\text{error}}}$$

Linearization

Let $x_0 \in \mathbb{R}^n$ be given point.

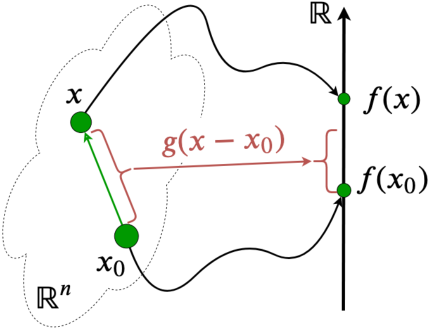
$$\underbrace{f(x) - f(x_0)}_{\text{change in function}} = \underbrace{g(x - x_0)}_{\substack{\text{linear part} \\ \text{(differential)}}} + \underbrace{h(x - x_0)}_{\substack{\text{approximation} \\ \text{error}}}$$



Linearization

Let $x_0 \in \mathbb{R}^n$ be given point.

$$\underbrace{f(x) - f(x_0)}_{\text{change in function}} = \underbrace{g(x - x_0)}_{\substack{\text{linear part} \\ \text{(differential)}}} + \underbrace{h(x - x_0)}_{\substack{\text{approximation} \\ \text{error}}}$$



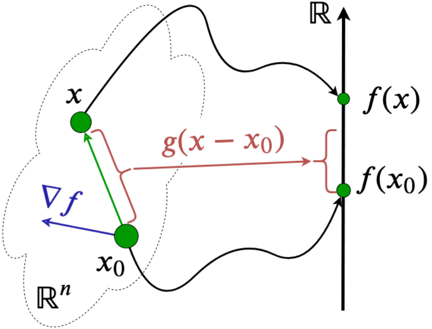
Linearization

Let $x_0 \in \mathbb{R}^n$ be given point.

$$\underbrace{f(x) - f(x_0)}_{\text{change in function}} = \underbrace{g(x - x_0)}_{\substack{\text{linear part} \\ \text{(differential)}}} + \underbrace{h(x - x_0)}_{\substack{\text{approximation} \\ \text{error}}}$$

Using Riesz theorem:
for some $\nabla f \in \mathbb{R}^n$ called gradient:

$$g(x - x_0) = \langle x - x_0, \nabla f \rangle$$



Descent

For some class of functions f («differentiable») we can say something about approximation error $h(x - x_0)$.

Descent

For some class of functions f («differentiable») we can say something about approximation error $h(x - x_0)$.

Consider some direction $x = x_0 + \alpha d$, $\alpha \in \mathbb{R}, d \in \mathbb{R}^n$:

$$f(x_0 + \alpha d) - f(x_0) = \alpha \langle d, \nabla f \rangle + h(\alpha d)$$

Descent

For some class of functions f («differentiable») we can say something about approximation error $h(x - x_0)$.

Consider some direction $x = x_0 + \alpha d$, $\alpha \in \mathbb{R}, d \in \mathbb{R}^n$:

$$f(x_0 + \alpha d) - f(x_0) = \alpha \langle d, \nabla f \rangle + h(\alpha d)$$

Using some 1d calculus:

$$\lim_{\alpha \to 0} \frac{\alpha \langle d, \nabla f \rangle + h(\alpha d)}{\alpha} = \langle d, \nabla f \rangle + \lim_{\alpha \to 0} \frac{h(\alpha d)}{\alpha}$$

Descent

For some class of functions f («differentiable») we can say something about approximation error $h(x - x_0)$.

Consider some direction $x = x_0 + \alpha d$, $\alpha \in \mathbb{R}, d \in \mathbb{R}^n$:

$$f(x_0 + \alpha d) - f(x_0) = \alpha \langle d, \nabla f \rangle + h(\alpha d)$$

Using some 1d calculus:

$$\begin{aligned} \lim_{\alpha \to 0} \frac{\alpha \langle d, \nabla f \rangle + h(\alpha d)}{\alpha} &= \langle d, \nabla f \rangle + \lim_{\alpha \to 0} \frac{h(\alpha d)}{\alpha} = \\ = \{\text{1d Taylor theorem}\} &= \langle d, \nabla f \rangle + 0 = \langle d, \nabla f \rangle \end{aligned}$$

Descent

For some class of functions f («differentiable») we can say something about approximation error $h(x - x_0)$.

Consider some direction $x = x_0 + \alpha d$, $\alpha \in \mathbb{R}, d \in \mathbb{R}^n$:

$$f(x_0 + \alpha d) - f(x_0) = \alpha \langle d, \nabla f \rangle + h(\alpha d)$$

Using some 1d calculus:

$$\begin{aligned} \lim_{\alpha \to 0} \frac{\alpha \langle d, \nabla f \rangle + h(\alpha d)}{\alpha} &= \langle d, \nabla f \rangle + \lim_{\alpha \to 0} \frac{h(\alpha d)}{\alpha} = \\ = \{\text{1d Taylor theorem}\} &= \langle d, \nabla f \rangle + 0 = \langle d, \nabla f \rangle \end{aligned}$$

if $\langle d, \nabla f \rangle < 0$, there is $\alpha > 0$:
$$f(x_0 + \alpha d) - f(x_0) < 0$$

Gradient Descent

How to choose direction d?

$$\begin{cases} g(x - x_0) \to \min\limits_x \\ \rho(x, x_0) \leq \varepsilon \end{cases}$$

Gradient Descent

How to choose direction d?

$$\begin{cases} g(x - x_0) \to \min\limits_{x} \\ \rho(x, x_0) \le \varepsilon \end{cases} \quad \Leftarrow \text{intuition: «trust region»}$$

Gradient Descent

How to choose direction d?

$$\begin{cases} g(x - x_0) \to \min\limits_{x} \\ \rho(x, x_0) \le \varepsilon \end{cases} \Leftarrow \text{intuition: «trust region»}$$

Standard choice of ρ: $\rho(x, x_0) := \sqrt{\langle x - x_0, x - x_0 \rangle}$

Gradient Descent

How to choose direction d?

$$\begin{cases} g(x - x_0) \to \min\limits_x \\ \rho(x, x_0) \le \varepsilon & \Leftarrow \text{intuition: «trust region»} \end{cases}$$

Standard choice of ρ: $\rho(x, x_0) := \sqrt{\langle x - x_0, x - x_0 \rangle}$
<u>**Solution:**</u> $x - x_0 \propto -\nabla f$

Generalization

Consider $f : \mathbb{R}^n \to \mathbb{R}^m$.

Generalization

Consider $f : \mathbb{R}^n \to \mathbb{R}^m$.

$$\begin{cases} f_1(x) - f_1(x_0) = g_1(x - x_0) + h_1(x - x_0) \\ f_2(x) - f_2(x_0) = g_2(x - x_0) + h_2(x - x_0) \\ \vdots \\ f_m(x) - f_m(x_0) = g_m(x - x_0) + h_m(x - x_0) \end{cases}$$

where all g_i are linear.

Generalization

Consider $f : \mathbb{R}^n \to \mathbb{R}^m$.

$$\begin{cases} f_1(x) - f_1(x_0) = g_1(x - x_0) + h_1(x - x_0) \\ f_2(x) - f_2(x_0) = g_2(x - x_0) + h_2(x - x_0) \\ \vdots \\ f_m(x) - f_m(x_0) = g_m(x - x_0) + h_m(x - x_0) \end{cases}$$

where all g_i are linear.

Just m different functions $\mathbb{R}^n \to \mathbb{R}$!

Generalization

Consider $f : \mathbb{R}^n \to \mathbb{R}^m$.

$$\begin{cases} f_1(x) - f_1(x_0) = \langle x - x_0, \nabla f_1 \rangle + h_1(x - x_0) \\ f_2(x) - f_2(x_0) = \langle x - x_0, \nabla f_2 \rangle + h_2(x - x_0) \\ \vdots \\ f_m(x) - f_m(x_0) = \langle x - x_0, \nabla f_m \rangle + h_m(x - x_0) \end{cases}$$

where all g_i are linear.

Just m different functions $\mathbb{R}^n \to \mathbb{R}$!

Generalization

Consider $f : \mathbb{R}^n \to \mathbb{R}^m$.

$$\begin{cases} f_1(x) - f_1(x_0) = \langle x - x_0, \nabla f_1 \rangle + h_1(x - x_0) \\ f_2(x) - f_2(x_0) = \langle x - x_0, \nabla f_2 \rangle + h_2(x - x_0) \\ \vdots \\ f_m(x) - f_m(x_0) = \langle x - x_0, \nabla f_m \rangle + h_m(x - x_0) \end{cases}$$

where all g_i are linear.

Just m different functions $\mathbb{R}^n \to \mathbb{R}$!

Corollary

All linear functions $\mathbb{R}^n \to \mathbb{R}^m$ are

$$g(x) = Ax$$

Jacobian

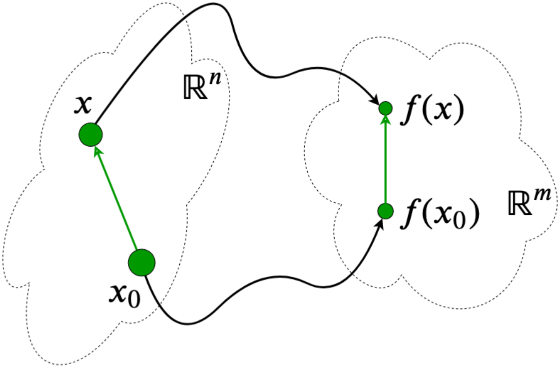
Define by $\nabla f \in \mathbb{R}^{m \times n}$ a matrix of component gradients:

$$f(x) - f(x_0) = \underbrace{\nabla f \cdot (x - x_0)}_{\substack{Df[x - x_0] \\ \text{differential}}} + h(x - x_0)$$

Jacobian

Define by $\nabla f \in \mathbb{R}^{m \times n}$ a matrix of component gradients:

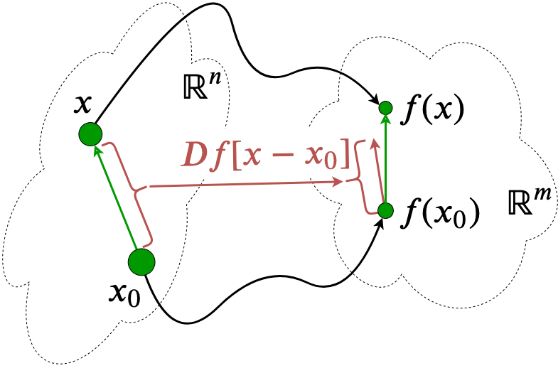
$$f(x) - f(x_0) = \underbrace{\nabla f \cdot (x - x_0)}_{\substack{Df[x - x_0] \\ \text{differential}}} + h(x - x_0)$$



Jacobian

Define by $\nabla f \in \mathbb{R}^{m \times n}$ a matrix of component gradients:

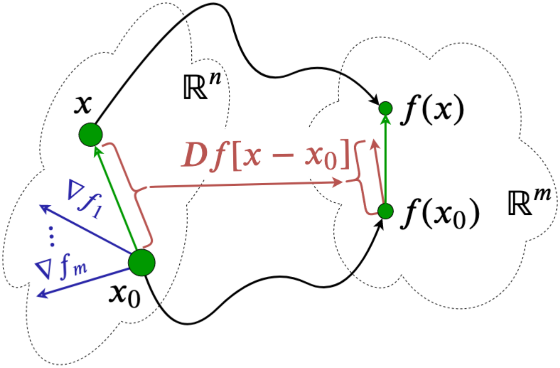
$$f(x) - f(x_0) = \underbrace{\nabla f \cdot (x - x_0)}_{\substack{Df[x - x_0] \\ \text{differential}}} + h(x - x_0)$$



Jacobian

Define by $\nabla f \in \mathbb{R}^{m \times n}$ a matrix of component gradients:

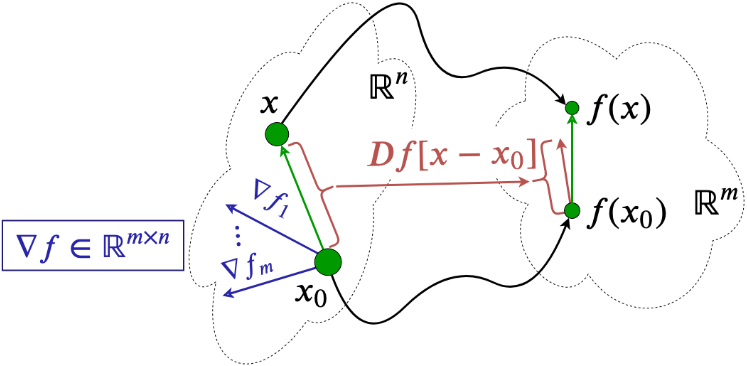
$$f(x) - f(x_0) = \underbrace{\nabla f \cdot (x - x_0)}_{\substack{Df[x - x_0] \\ \text{differential}}} + h(x - x_0)$$



Jacobian

Define by $\nabla f \in \mathbb{R}^{m \times n}$ a matrix of component gradients:

$$f(x) - f(x_0) = \underbrace{\nabla f \cdot (x - x_0)}_{\substack{Df[x-x_0] \\ \text{differential}}} + h(x - x_0)$$



Comparing jacobian and differential

	jacobian	differential
Dimensions	$m \times n$	m
Depends on	x_0	$x_0, x - x_0$

Comparing jacobian and differential

	jacobian	differential
Dimensions	$m \times n$	m
Depends on	x_0	$x_0, x - x_0$

Question: what to do if argument or value of function is matrix?

Comparing jacobian and differential

	jacobian	differential
Dimensions	$m \times n$	m
Depends on	x_0	$x_0, x - x_0$

Question: what to do if argument or value of function is matrix?

$$\mathbb{R}^{n \times m} \cong \mathbb{R}^{nm}$$

Comparing jacobian and differential

	jacobian	differential
Dimensions	$m \times n$	m
Depends on	x_0	$x_0, x - x_0$

Question: what to do if argument or value of function is matrix?

$$\mathbb{R}^{n \times m} \cong \mathbb{R}^{nm}$$

Corollary

Let $A, B \in \mathbb{R}^{n \times m}$

$$\langle A, B \rangle_{\mathbb{R}^{n \times m}} = \langle A.\text{flatten}(), B.\text{flatten}() \rangle_{\mathbb{R}^{nm}}$$

Comparing jacobian and differential

	jacobian	**differential**
Dimensions	$m \times n$	m
Depends on	x_0	$x_0, x - x_0$

Question: what to do if argument or value of function is matrix?

$$\mathbb{R}^{n \times m} \cong \mathbb{R}^{nm}$$

Corollary

Let $A, B \in \mathbb{R}^{n \times m}$

$$\langle A, B \rangle_{\mathbb{R}^{n \times m}} = \langle A.\text{flatten}(), B.\text{flatten}() \rangle_{\mathbb{R}^{nm}} = tr(B^T A)$$

Backpropagation

Vector differentiation

Constructing complex functions

Questions:

- what functions $\mathbb{R}^n \to \mathbb{R}$ do we know?

Constructing complex functions

Questions:

- what functions $\mathbb{R}^n \to \mathbb{R}$ do we know?
- how to automatically calculate their gradient?

Constructing complex functions

Questions:

- what functions $\mathbb{R}^n \to \mathbb{R}$ do we know?
- how to automatically calculate their gradient?

1. find some primitive building blocks $\mathbb{R}^n \to \mathbb{R}^m$

Constructing complex functions

Questions:

- what functions $\mathbb{R}^n \to \mathbb{R}$ do we know?
- how to automatically calculate their gradient?

1. find some primitive building blocks $\mathbb{R}^n \to \mathbb{R}^m$
2. find their jacobians/differentials analytically.

Constructing complex functions

Questions:

- what functions $\mathbb{R}^n \to \mathbb{R}$ do we know?
- how to automatically calculate their gradient?

1. find some primitive building blocks $\mathbb{R}^n \to \mathbb{R}^m$
2. find their jacobians/differentials analytically.
3. construct complex functions using composition

Constructing complex functions

Questions:

- what functions $\mathbb{R}^n \to \mathbb{R}$ do we know?
- how to automatically calculate their gradient?

1. find some primitive building blocks $\mathbb{R}^n \to \mathbb{R}^m$
2. find their jacobians/differentials analytically.
3. construct complex functions using composition
4. apply chain rule!

Building blocks

Let $x, y \in \mathbb{R}^n$ be input vector.

- element-wise application ("map") of some scalar function.
  - **examples:** e^x, x^2, $x + 1$, $\frac{1}{x}$...

Building blocks

Let $x, y \in \mathbb{R}^n$ be input vector.

- element-wise application ("map") of some scalar function.
  - **examples:** e^x, x^2, $x + 1$, $\frac{1}{x}$...
- element-wise operations
  - **examples:** $x + y$, $x * y$, $\frac{x}{y}$...

Building blocks

Let $x, y \in \mathbb{R}^n$ be input vector.

- element-wise application ("map") of some scalar function.
  - **examples:** e^x, x^2, $x+1$, $\frac{1}{x}$...
- element-wise operations
  - **examples:** $x+y$, $x*y$, $\frac{x}{y}$...
- scalar product
  - **examples:** $\langle x, y\rangle$, Ax

Building blocks

Let $x, y \in \mathbb{R}^n$ be input vector.

- element-wise application ("map") of some scalar function.
  - **examples:** e^x, x^2, $x+1$, $\frac{1}{x}$...
- element-wise operations
  - **examples:** $x+y$, $x*y$, $\frac{x}{y}$...
- scalar product
  - **examples:** $\langle x, y \rangle$, Ax
- accumulating ("reduce") operations
  - **examples:** sum/max/min of all components

Building blocks

Let $x, y \in \mathbb{R}^n$ be input vector.

- element-wise application ("map") of some scalar function.
  - **examples:** e^x, x^2, $x+1$, $\frac{1}{x}$...
- element-wise operations
  - **examples:** $x+y$, $x*y$, $\frac{x}{y}$...
- scalar product
  - **examples:** $\langle x, y \rangle$, Ax
- accumulating ("reduce") operations
  - **examples:** sum/max/min of all components
- something special
  - **examples:** matrix inverse

Chain Rule: setting

Given:

$$y(x) : \mathbb{R}^n \to \mathbb{R}^m \text{ with jacobian } \nabla_x y \in \mathbb{R}^{m \times n} \text{ at point } x_0$$
$$z(y) : \mathbb{R}^m \to \mathbb{R}^k \text{ with jacobian } \nabla_y z \in \mathbb{R}^{k \times m} \text{ at point } y_0$$

Chain Rule: setting

Given:

$$y(x) : \mathbb{R}^n \to \mathbb{R}^m \text{ with jacobian } \nabla_x y \in \mathbb{R}^{m \times n} \text{ at point } x_0$$
$$z(y) : \mathbb{R}^m \to \mathbb{R}^k \text{ with jacobian } \nabla_y z \in \mathbb{R}^{k \times m} \text{ at point } y_0$$

the task is to find jacobian $\nabla_x z \in \mathbb{R}^{k \times n}$ of function

$$z(x) = z(y(x)) : \mathbb{R}^n \to \mathbb{R}^k$$

at point x_0.

Chain Rule: setting

Given:

$$y(x) : \mathbb{R}^n \to \mathbb{R}^m \text{ with jacobian } \nabla_x y \in \mathbb{R}^{m \times n} \text{ at point } x_0$$
$$z(y) : \mathbb{R}^m \to \mathbb{R}^k \text{ with jacobian } \nabla_y z \in \mathbb{R}^{k \times m} \text{ at point } y_0$$

the task is to find jacobian $\nabla_x z \in \mathbb{R}^{k \times n}$ of function

$$z(x) = z(y(x)) : \mathbb{R}^n \to \mathbb{R}^k$$

at point x_0.

Centralize everything:

$$\Delta x = x - x_0$$

$$\Delta y = y - y_0$$

$$\Delta z = z - z_0$$

Chain Rule for jacobians

$$\Delta y = \nabla_x y \Delta x + \bar{\bar{o}}(\Delta x)$$

$$\Delta z = \nabla_y z \Delta y + \bar{\bar{o}}(\Delta y)$$

$$\Delta z = \nabla_x z \Delta x + \bar{\bar{o}}(\Delta x)$$

Chain Rule for jacobians

$$\Delta y = \nabla_x y \Delta x + \bar{\bar{o}}(\Delta x)$$

$$\Delta z = \nabla_y z \Delta y + \bar{\bar{o}}(\Delta y)$$

$$\Delta z = \nabla_x z \Delta x + \bar{\bar{o}}(\Delta x)$$

Insert first in second:

$$\Delta z = \nabla_y z \nabla_x y \Delta x + \nabla_y z \bar{\bar{o}}(\Delta x) + \bar{\bar{o}}(\nabla_x y \Delta x + \bar{\bar{o}}(\Delta x))$$

Chain Rule for jacobians

$$\Delta y = \nabla_x y \Delta x + \bar{\bar{o}}(\Delta x)$$
$$\Delta z = \nabla_y z \Delta y + \bar{\bar{o}}(\Delta y)$$
$$\Delta z = \color{red}{\nabla_x z} \Delta x + \bar{\bar{o}}(\Delta x)$$

Insert first in second:

$$\Delta z = \nabla_y z \nabla_x y \Delta x + \nabla_y z \bar{\bar{o}}(\Delta x) + \bar{\bar{o}}(\nabla_x y \Delta x + \bar{\bar{o}}(\Delta x)) = \\ = \nabla_y z \nabla_x y \Delta x + \bar{\bar{o}}(\Delta x)$$

Chain Rule for jacobians

$$\Delta y = \nabla_x y \Delta x + \bar{\bar{o}}(\Delta x)$$

$$\Delta z = \nabla_y z \Delta y + \bar{\bar{o}}(\Delta y)$$

$$\Delta z = \nabla_x z \Delta x + \bar{\bar{o}}(\Delta x)$$

Insert first in second:

$$\Delta z = \nabla_y z \nabla_x y \Delta x + \nabla_y z \bar{\bar{o}}(\Delta x) + \bar{\bar{o}}(\nabla_x y \Delta x + \bar{\bar{o}}(\Delta x)) = \\ = \nabla_y z \nabla_x y \Delta x + \bar{\bar{o}}(\Delta x)$$

Chain rule for jacobians

$$\nabla_x z = \nabla_y z \nabla_x y$$

Chain Rule for differentials

$$\Delta y = D_x y[\Delta x] + \bar{\bar{o}}(\Delta x)$$

$$\Delta z = D_y z[\Delta y] + \bar{\bar{o}}(\Delta y)$$

$$\Delta z = D_x z[\Delta x] + \bar{\bar{o}}(\Delta x)$$

Chain Rule for differentials

$$\Delta y = D_x y[\Delta x] + \bar{\bar{o}}(\Delta x)$$
$$\Delta z = D_y z[\Delta y] + \bar{\bar{o}}(\Delta y)$$
$$\Delta z = D_x z[\Delta x] + \bar{\bar{o}}(\Delta x)$$

Insert first in second:

$$\Delta z = D_y z[D_x y[\Delta x]] + D_y z[\bar{\bar{o}}(\Delta x)] + \bar{\bar{o}}(D_x y[\Delta x] + \bar{\bar{o}}(\Delta x))$$

Chain Rule for differentials

$$\Delta y = D_x y[\Delta x] + \bar{\bar{o}}(\Delta x)$$

$$\Delta z = D_y z[\Delta y] + \bar{\bar{o}}(\Delta y)$$

$$\Delta z = D_x z[\Delta x] + \bar{\bar{o}}(\Delta x)$$

Insert first in second:

$$\Delta z = D_y z[D_x y[\Delta x]] + D_y z[\bar{\bar{o}}(\Delta x)] + \bar{\bar{o}}(D_x y[\Delta x] + \bar{\bar{o}}(\Delta x)) =$$
$$= D_y z[D_x y[\Delta x]] + \bar{\bar{o}}(\Delta x)$$

Chain Rule for differentials

$$\Delta y = D_x y[\Delta x] + \bar{\bar{o}}(\Delta x)$$
$$\Delta z = D_y z[\Delta y] + \bar{\bar{o}}(\Delta y)$$
$$\Delta z = D_x z[\Delta x] + \bar{\bar{o}}(\Delta x)$$

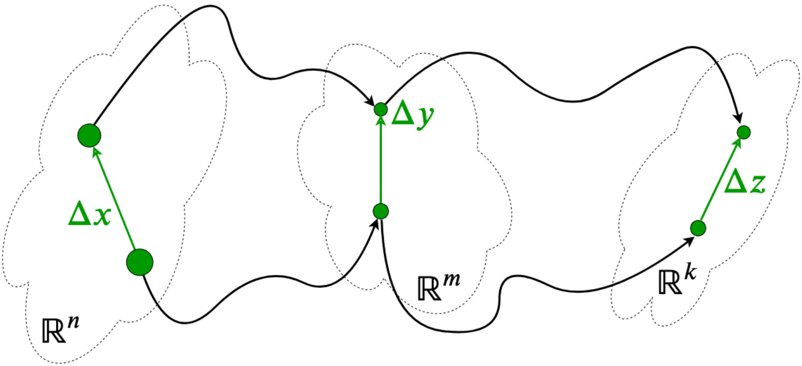
Insert first in second:

$$\Delta z = D_y z[D_x y[\Delta x]] + D_y z[\bar{\bar{o}}(\Delta x)] + \bar{\bar{o}}(D_x y[\Delta x] + \bar{\bar{o}}(\Delta x)) = \\ = D_y z[D_x y[\Delta x]] + \bar{\bar{o}}(\Delta x)$$

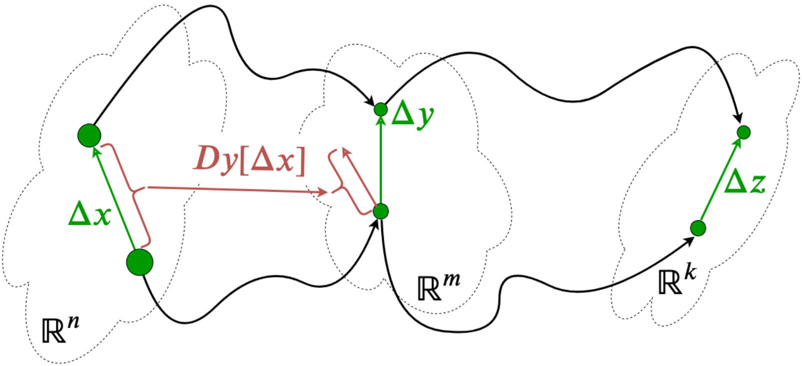
Chain rule for differentials

$$D_x z[\Delta x] = D_y z[D_x y[\Delta x]]$$

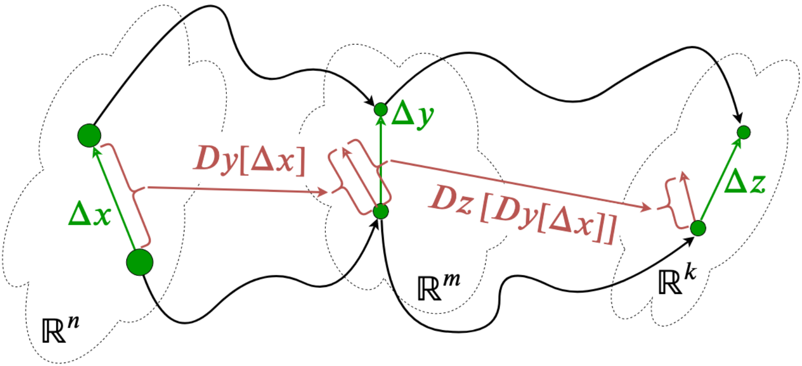
Chain Rule intuition



Chain Rule intuition



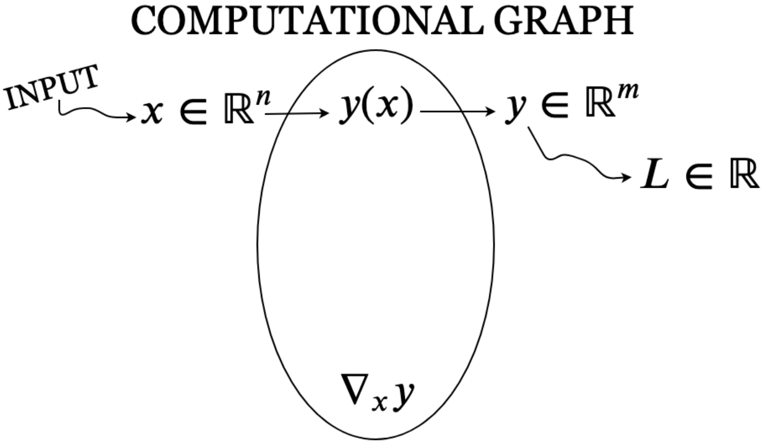
Chain Rule intuition



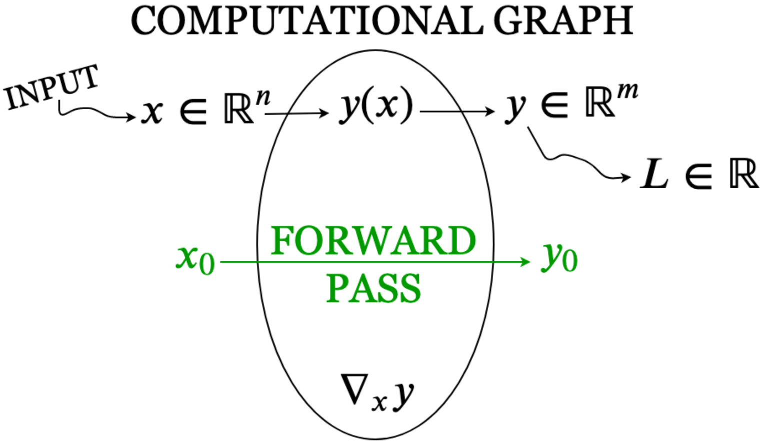
Backpropagation

Backpropagation

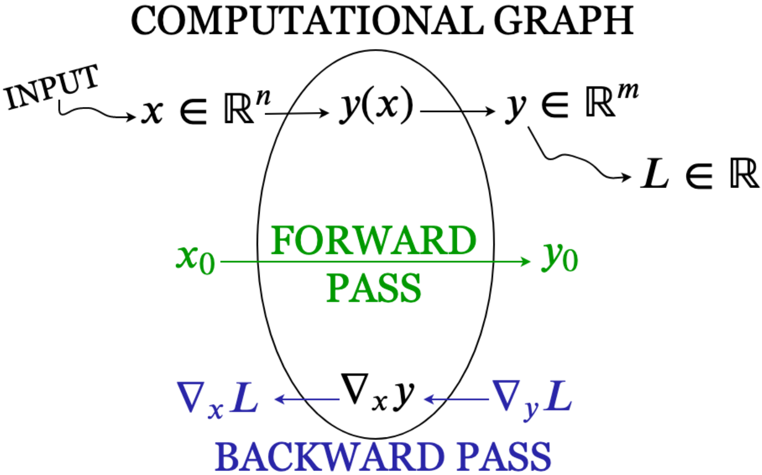
Automatic differentiation



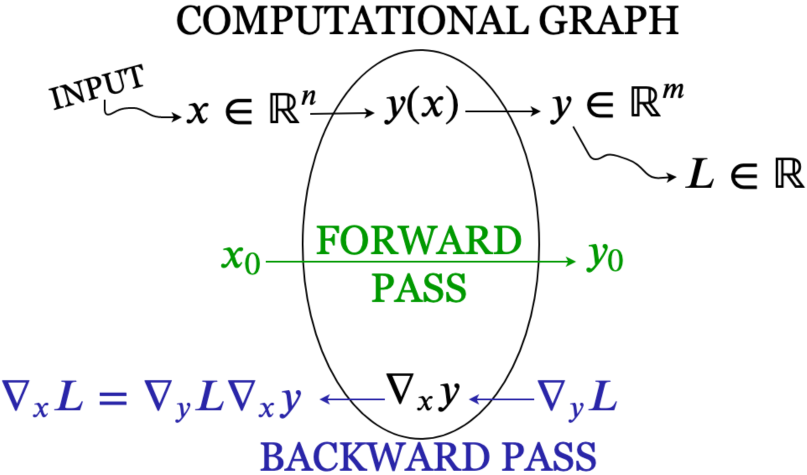
Automatic differentiation



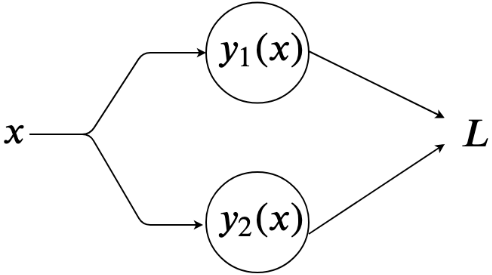
Automatic differentiation



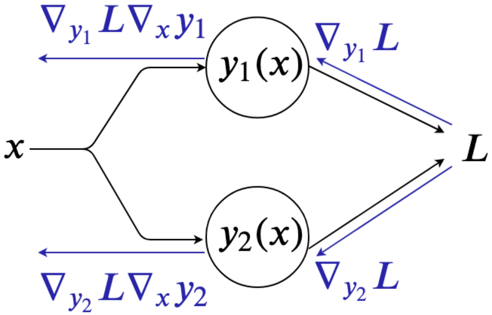
Automatic differentiation



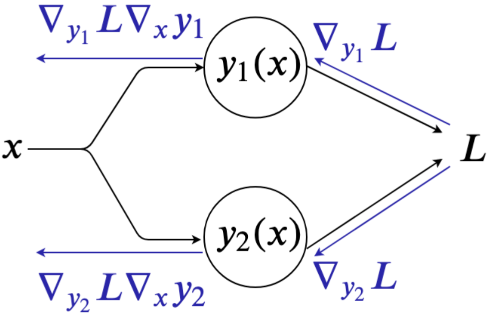
Parallel computations



Parallel computations



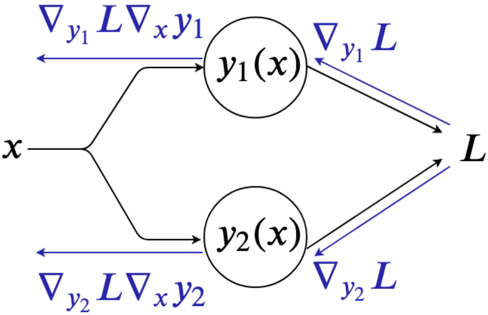
Parallel computations



Let $y = [y_1, y_2]$:

$$\Delta L = \nabla_y L \Delta y + \bar{\bar{o}} =$$

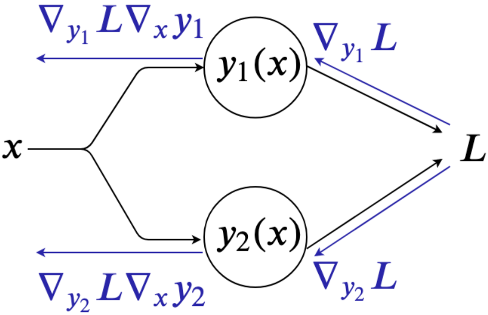
Parallel computations



Let $y = [y_1, y_2]$:

$$\Delta L = \nabla_y L \Delta y + \bar{\bar{o}} = \nabla_{y_1} L \Delta y_1 + \nabla_{y_2} L \Delta y_2 + \bar{\bar{o}}$$

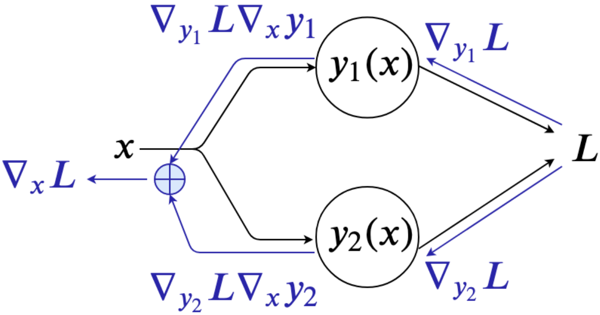
Parallel computations



Let $y = [y_1, y_2]$:

$$\Delta L = \nabla_y L \Delta y + \bar{\bar{o}} = \nabla_{y_1} L \Delta y_1 + \nabla_{y_2} L \Delta y_2 + \bar{\bar{o}} =$$
$$= \nabla_{y_1} L \nabla_x y_1 \Delta x + \nabla_{y_2} L \nabla_x y_2 \Delta x + \bar{\bar{o}}$$

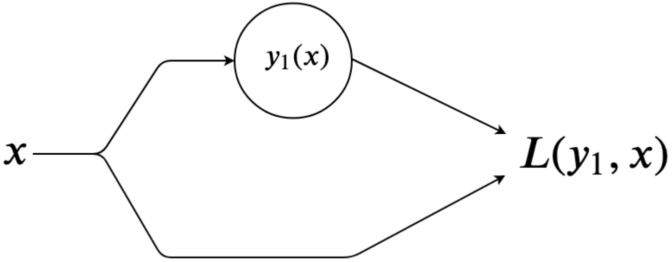
Parallel computations



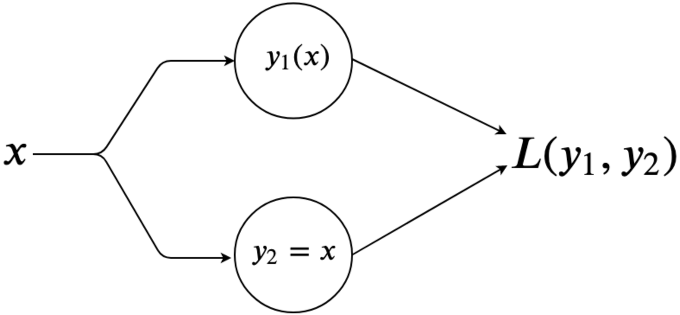
Let $y = [y_1, y_2]$:

$$\begin{aligned}\Delta L = \nabla_y L \Delta y + \bar{\bar{o}} = \nabla_{y_1} L \Delta y_1 + \nabla_{y_2} L \Delta y_2 + \bar{\bar{o}} = \\ = \nabla_{y_1} L \nabla_x y_1 \Delta x + \nabla_{y_2} L \nabla_x y_2 \Delta x + \bar{\bar{o}} = \\ = \left(\nabla_{y_1} L \nabla_x y_1 + \nabla_{y_2} L \nabla_x y_2 \right) \Delta x + \bar{\bar{o}}\end{aligned}$$

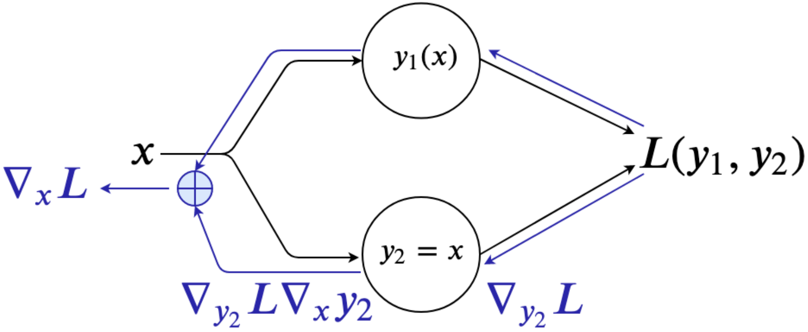
Arbitrary graphs



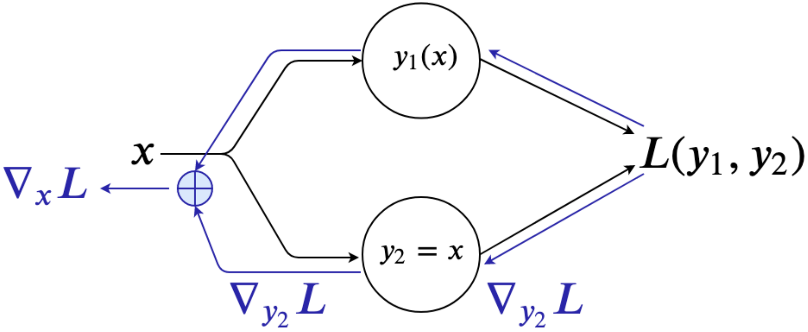
Arbitrary graphs



Arbitrary graphs



Arbitrary graphs



Arbitrary graphs